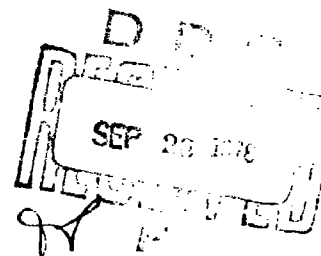ESD-TR-78-158

LEVEL

MTR-3592 Vol. 1

LIMITATIONS OF END-TO-END ENCRYPTION
IN SECURE COMPUTER NETWORKS

BY M.A. PADLIPSKY, D.W. SNOW, P.A. KARGER

AUGUST 1978

Prepared for

DEPUTY FOR TECHNICAL OPERATIONS
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
Hanscom Air Force Base, Massachusetts

SEP 28 1978

REVIEW AND APPROVAL

This technical report has been reviewed and is approved for publication.

WILLIAM R. PRICE, Captain, USAF
Technology Applications Division

CHARLES J. GREWE, Jr., Lt Col, USAF
Chief, Technology Applications Division

FOR THE COMMANDER

RICHARD P. RUBRECHT, Lt Colonel, USAF
Acting Director, Computer Systems Engineering
Deputy for Technical Operations
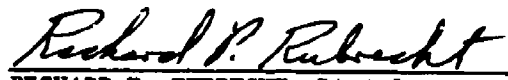
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ESD-TR-78-158	**2. GOVT ACCESSION NO.**	**3. RECIPIENT'S CATALOG NUMBER**
4. TITLE (and Subtitle) LIMITATIONS OF END-TO-END ENCRYPTION IN SECURE COMPUTER NETWORKS		**5. TYPE OF REPORT & PERIOD COVERED**
		6. PERFORMING ORG. REPORT NUMBER MTR-3592, Vol. I
7. AUTHOR(s) M. A. Padlipsky, D. W. Snow and P. A. Karger		**8. CONTRACT OR GRANT NUMBER(s)** F19628-78-C-0001
9. PERFORMING ORGANIZATION NAME AND ADDRESS The MITRE Corporation P.O. Box 208 Bedford, MA 01730		**10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS** Project No. 672B
11. CONTROLLING OFFICE NAME AND ADDRESS Deputy for Technical Operations Electronic Systems Division, AFSC Hanscom Air Force Base, MA 01731		**12. REPORT DATE** AUGUST 1978
		13. NUMBER OF PAGES 12
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		**15. SECURITY CLASS. (of this report)** UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

COMMUNICATIONS SECURITY
COMPUTER SECURITY
ENCRYPTION
NETWORK SECURITY

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

End-to-end encryption is not sufficient to prevent compromise of information in a network that employs untrusted subnet (network interface) processors. Addresses, message lengths, and timing of transmissions furnish channels whose bandwidth can be significant.

DD FORM 1473 EDITION OF 1 NOV 65 IS OBSOLETE
1 JAN 73

ACKNOWLEDGMENT

TABLE OF CONTENTS

INTRODUCTION

In a recent paper on computer network security, the claim is advanced that "network cryptographic devices (of the special kind described therein) virtually eliminate security threats to network communications..."* This rather strong view appears to typify a sizeable segment of opinion in the field; indeed, informal conversations occasionally give the impression that some believe that cryptography can guarantee security. However, it is our contention that cryptography -- and "end-to-end encryption" (described below) in particular -- is far from being a panacea. As we shall show, a computer network which relies on end-to-end encryption to avoid the necessity of developing trustworthy Hosts and/or communications subnetwork processors (CSNPs) is vulnerable to compromise in several ways.

SENDERS AND RECEIVERS

There are two major kinds of encryption which may, separately or jointly, be employed in computer networks: link encryption and end-to-end encryption. Schematically,

where the Hosts are the computers using the network to communicate, the CSNPs are the communications subnetwork processors, the LEHs are link encryption hardware which encrypt on a point-to-point basis all information being transmitted through the communication medium, and the EEHs are end-to-end encryption hardware which encrypt all data being sent from the Host to the CSNP (but may pass some control information, such as the address a given message is to be sent to, unencrypted). Although the value and necessity of link encryption in defeating "wire tapping" of the medium and preventing traffic flow analysis by concealing addresses are clear, end-to-end encryption in a computer network is not equally effective.

---

* Heinrick, Frank R. and David J. Kaufman, "A Centralized Approach to Computer Network Security", AFIPS Conference Proceedings, Volume 45, pp. 85-90, AFIPS Press, 1976.

Assume that some unspecified type of end-to-end encryption hardware is inserted between a Host and an untrusted* CSNP in order to prevent the CSNP from being able to read the classified data to be transmitted through it. (Decryption will occur between the destination CSNP and its Host; we will not address the handling of cryptographic "keys" other than to postulate that it can be achieved.) Assume further that the network itself has Hosts at more than one security level. For compromise to occur, there must be a sender of classified information within the Host, one or more information channels, and a receiver outside the Host. Our first concern will be identifying potential senders and receivers.

From the fact that we have made no assumptions about the trustedness of the Host software, it follows that the sort of "Trojan Horse" programs of the computer security literature could be present:

> This rather interesting attack is directed to placing code with trap doors into a target system. It attempts to achieve this by presenting the operators of the system with a program so useful that they will use it even though it may not have been produced under their control. An ideal 'gift' of this kind would be a text editor or other major system function that requires access to user files as part of the function. If the Trojan Horse routine opens the user files for him as part of the 'service', the program also has the opportunity to record the user ID and/or passwords on his file. It may also be possible to copy all or part of the file being 'edited' to a file accessible to a penetrator.**

---

* By "untrusted" we mean to convey that the software has neither been verified nor validated to be correct. The term "uncertified" is sometimes applied to such situations, but is avoided here because certification is properly an administrative action, which can be performed in the absence of formal verification or validation.

** J. P. Anderson, "Computer Security Technology Planning Study", ESD-TR-73-51, Vols. I and II, James P. Anderson and Company, Fort Washington, Pa., October 1972 (AD 758206 and AD 772806).

Such Trojan horses, then, clearly could serve as senders of classified information from the Host, provided they have channels to send the information over.

Before turning to potential receivers of the classified information, two important points should be observed about Trojan horses: (1) They do not require "live user" intervention to be actuated. Rather, they can be programmed to transmit either all the time, or in the response to some indication that the receiver is ready, or at particular times of day -- or, indeed, under any programmable circumstances. Thus, appeal to the contention that only cleared users operate a system's software is irrelevant, always assuming that the Trojan horses have information channels available. (2) The reason why we single out the Trojan horse threat in particular from the several kinds of flaws, cited in the literature, that could be present in Host software is that they could be present and act as potential senders in any system in which it is not the case that all software has been verified (which is equivalent to saying in any known system).

Security kernel technology has been proposed* as a defense against Trojan horses, as well as accidental flaws. A security kernel is that part of an operating system whose correctness is both critical and sufficient to ensure data protection even when the rest of the operating system and other software is untrusted. A security kernel does not suppress Trojan horses; it merely prevents them from compromising information within the Host. Information sent out to a CSNP, however, is out of Host security kernel control.

With Trojan horse programs as potential senders of classified information, then, the next problem is to identify potential receivers of the information: cooperating Trojan horses in the CSNPs are an obvious candidate, of course. For that matter, if the CSNPs of a given network are not all physically secure, a CSNP could be penetrated at any point in time, without ever having to go to the trouble of implanting a Trojan horse during development. It is also possible for literal or figurative wire-tapping to occur -- depending, of course, on the physical security of a given network -- at any point (other than the CSNP-CSNP communications medium if link encryption is also employed). Note that should the receiver be a Trojan horse in the by-hypothesis untrusted CSNP, it would then essentially be at liberty to use the network itself to pass the information along to a human confederate at any uncleared Host or terminal on the net.

---

*See, for example, D.E. Bell and L.J. LaPadula, "Secure Computer System: Unified Exposition and Multics Interpretation", ESD-TR-75-306, Electronic Systems Division, AFSC, Hanscom AFB, MA, July 1975 (AD A023588).

CHANNELS

The possible existence of senders and receivers is, of course, no threat if they have no means of communicating. The question then becomes one of finding communication channels despite the presence of end-to-end encryption hardware (EEH). We suggest three such channels can exist:

The key to the first channel is the observation that, although EEH conceals the contents of network transmissions from the potential receivers (either in the CSNP itself, or tapping "wires"), it cannot conceal the address of the transmissions.* So, if there is more than one destination possible for a given Host to transmit to via a single EEH-CSNP pair, the of-necessity unencrypted "field" (explicit or implicit) containing the address is available as a channel between illicit senders and receivers. ("Bandwidth" for all three channels will be discussed below.) It might seem that an EEH-CSNP pair could be employed for each remote Host that a given Host is permitted to communicate with on a classified level, but there are several problems with such a countermeasure: In the first place, it could be prohibitively expensive in networks of reasonable size. Second, if the sending code resides in the Host's Network Control Program, the order of transmissions to the separate EEH-CSNP pairs could be modulated (and the CSNPs, being on the same network, can communicate freely with one another), giving the same effect as if an explicit address field were modulated. Finally, multi-CSNP Hosts (which is what the situation would "look like" to the network) can lead to awkward network software protocols.

The key to the second channel is the observation that the lengths of transmissions from the Host to the CSNP are likely to be observable by the cited receivers, and could also be modulated by the cited senders in order to pass illicitly acquired information. The EEH could, of course, pad all transmissions up to the given network's maximum message size with blanks (which, once encrypted, would not be recognizable by the receiver); but this can be expensive in terms of real bandwidth, as the dummy bits must traverse the network taking up resources that would otherwise be available for real bits. Even the expedient of padding up to the nearest

---

* We assume that the EEH performs whatever Host-CSNP protocol is necessary, so that a potentially penetrated Network Control Program in a Host cannot communicate directly with a potentially penetrated CSNP at will, via fictitious Host-CSNP commands. (Note that this prevents Hosts setting priorities for particular messages.)

7

convenient fraction of the maximum (e.g., appearing to transmit in
512 or 1024 bit increments) is not sufficient to close the length
channel, although it does decrease the available bandwidth (see
below).

Finally, even if a given network found it acceptable to do
only Host pair-wise transmission at the maximum message size in
order to defeat the address and length channels, it is still the case
that the very fact of transmission is observable by the cited re-
ceivers. Thus, a timing*channel exists which can only be countered
by having the EEH "always" appear to be transmitting. That is, if
legitimate traffic is available, the EEH transmits it; if not, it
sends dummy traffic, up to the maximum rate the CSNP can handle traffic
(or at some fixed, lesser rate). Such a tactic would, however,
also decrease real bandwidth - either by keeping the communications
subnet too full of dummy traffic, or by refusing to service peak
rate traffic as rapidly as it could be serviced.

It appears, then, that the alternative to permitting the cited
channels to exist is to impose constraints upon the network that,
to this point in the discussion, at least, might well be unacceptable.
Before exploring the topic of countermeasures further, however,
some attempt should be made to quantify the threat, for if the
channels are sufficiently "slow" they might be declared not to be
significant as a policy matter for given networks.

BANDWIDTH

How fast the cited channels can operate varies according to
the particular networks and Hosts at hand, of course. We can see,
however, that the upper bound on the address and length channels
taken separately (if every transmission is known to contain illicit
information) is

$$n \times w$$

where n is the number of transmissions per second possible to a
given CSNP from a given Host, and w is the width in bits of the
explicity or implicity address or length field. Thus, in a network
where the CSNP can handle a not unreasonable 10 transmissions per

* This is a "covert" channel as discussed by B. W. Lampson,
  "A Note on the Confinement Problem", Comm. ACM, Vol. 16,
  No. 10 (October 1973), pp. 613-615.

8

second from a Host,* a not uncommon 8-bit wide address field would allow 80 bits per second to be communicated "around" end-to-end encryption hardware which permitted all addresses to be sent from the Host to the CSNP, and an also not uncommon 10-bit length field, 100 bps (provided the Host-CSNP interface does in fact support the bit rate necessary to perform the 10 transmissions). As the standard military teletype operates at only 75 bps, the 80-100 bps rates should be sufficiently significant that it is almost un- necessary to observe the following: Assuming that the Host's Network Control Program is the sender (i.e., contains the Trojan Horse), not only can the receiver know that all transmissions (or all transmissions between pre-established marker values) are meaning- ful, but both the address field and the length field can be used together, giving 180 bps on the probably conservative assumptions above. The functional effect is that an uncleared user can be pro- vided essentially an interactive terminal on a classified computer system -- usable at his choice and without the knowledge of the classified facility.

Although recognition of the fact that a given transmission comes from a confederate in the Host is more difficult when the Network Control Program itself is not the confederate, communication over "noisy" channels is possible by use of redundancy. Therefore, rather than go through the exercise of inventing recognition schemes, let us accept the contention that the address and length channels constitute a noticeable threat and turn to the timing channel.** Here, it seems, the threat is far harder even to estimate, as the variabilities of actual CSNPs have strong impact on what level of timing discrimination they're capable of. As a rule of thumb, though, again given the assumption of a penetrated Network Control

---

* For minimum-length transmission, 10/sec might be an order of magnitude low, so the consideration that some of the time legitimate traffic must be sent instead of fictitious traffic (which is fabri- cated only to have address and/or length fields to modulate) does not detract materially from the thrust of the argument. And in those cases where the length is freely observable by the receiver, all transmissions can be used by a penetrated NCP, by virtue of repacking messages to get lengths to suit the needs of whatever code is being employed.

** Just to avert the suspicion of handwaving: recognition that a given transmission is from the Host-side confederate (and hence signifi- cant) could be achieved in many networks simply by using odd- numbered bit lengths (all of which would be significant) or by addressing a little-used Host (with length significant), to name but two.

9

Program, the presence or absence of a transmission during a discernable time slice would constitute a "bit", so the rate would depend strictly upon the number of minimum length transmissions sendable from the Host to the CSNP per second; 10-100 bps seems to be a reasonable range for a timing channel operated in such a fashion. We are not so confident that the timing channel could be operated at teletype rates as we were that the address and length channels could, but given the right circumstances it might well be able to.

COUNTERMEASURES

The foregoing suggests that it would be desirable to apply countermeasures to the cited channels. As argued earlier, however, end-to-end encryption hardware cannot do the job by itself without unpleasant cost-performance consequences. What of countermeasures to the cited senders or receivers?

Attempting to prevent the sending of illicitly-acquired information out of the Host is not too fruitful. Certifying the entire software complement of the Host is clearly too difficult an undertaking. An alternative might be to certify only the Network Control Program, placing it in a front-end machine for system-high Hosts, or making it part of the security kernel for multi-level secure Hosts, to prevent tampering. Then the certified NCP could block the length and timing channels, just as the EEH could; however, the performance effects would be as bad or worse with the blocking done in software, and it is by no means clear that software can block the address channel completely. (A certified NCP could prevent a given Host process/job/task from opening network connections to more than one host, but could do nothing to counter cooperating processes modulating the address field.) For practical purposes, the difficulty in recognizing which transmissions are significant when the NCP is not a potential confederate might slow the address channel down to an acceptable level, but blocking the potential senders is not a general solution.

The remaining area for countermeasures is that of the cited receivers, and this does appear to be a fruitful one. For if the receivers can either be eliminated or prevented from passing the illicitly-acquired information along, then a general solution can be said to have been achieved. Of the cited receivers, "wire" tappers can be eliminated by insisting on physical security on the Host-CSNP connection and link encryption (which would conceal the channels) on the CSNP-CSNP connections. What remains is the potential Trojan Horse within the CSNP; if it can be neutralized in some fashion, then information might leave the Host, but it could not be used.

10

One possibility is the complete certification of the code within the CSNP, if it is not too extensive and it is properly designed to be certifiable. Should even this expedient prove too difficult, however, there remains another alternative; let the CSNPs operating system be based on a security kernel. In particular, any demultiplexing of transmissions from the Host (necessary if the Host is multilevel secure, so as to de ermine at what security level the transmission is to be handled with the CSNP) is performed within the kernel. The effect, even for system high Hosts, is to associate a level with each transmission, indelibly. Then any routine processing, such as routing, can be performed by uncertified processes <u>operating at the level of the transmission</u> because such processes will only be permitted to send to the network (again, through the kernel) at the same level, by definition.* Thus, if a receiving Trojan Horse were present, it could not pass illicitly-acquired information along to a human agent -- even if the Host-CSNP transmissions were in clear text. So not only is end-to-end encryption not sufficient, but with the appropriate CSNP, and link encryption, it is not even necessary.

CONCLUSIONS

The introduction of end-to-end encryption hardware into a network that employs untrusted communications subnet processors has been shown to leave the network subject to security compromise because potential senders of classified information have several channels (addresses, lengths, and timing of transmissions) available through which to communicate with potential receivers. Although it is at best extremely difficult to eliminate the potential senders or to block the channels, it does seem that the potential software receivers of the information can be prevented from further communicating the information to human agents.** The security kernel-based communications subnetwork processor to do this, however, could even be permitted to receive unencrypted transmissions from the Host. Therefore, end-to-end encryption is neither sufficient to guarantee computer network security, nor is it necessary to achieve it.

---

* Withou* wandering too far afield into the details of security kernel technology, one way of viewing the key point is to note that a kernel-based Host was vulnerable to Trojan Horses because full control could not be exercised over output if the associated CSNP were not itself trusted but were able to communicate with Hosts at lower security levels; a kernel-based CSNP, however, does allow output to be controlled fully.

** By a combination of link encryption, physical security on all communications subnet processors, and security kernels in all CSNPs.